Title: Lecture - Machine Learning, PHYS 777

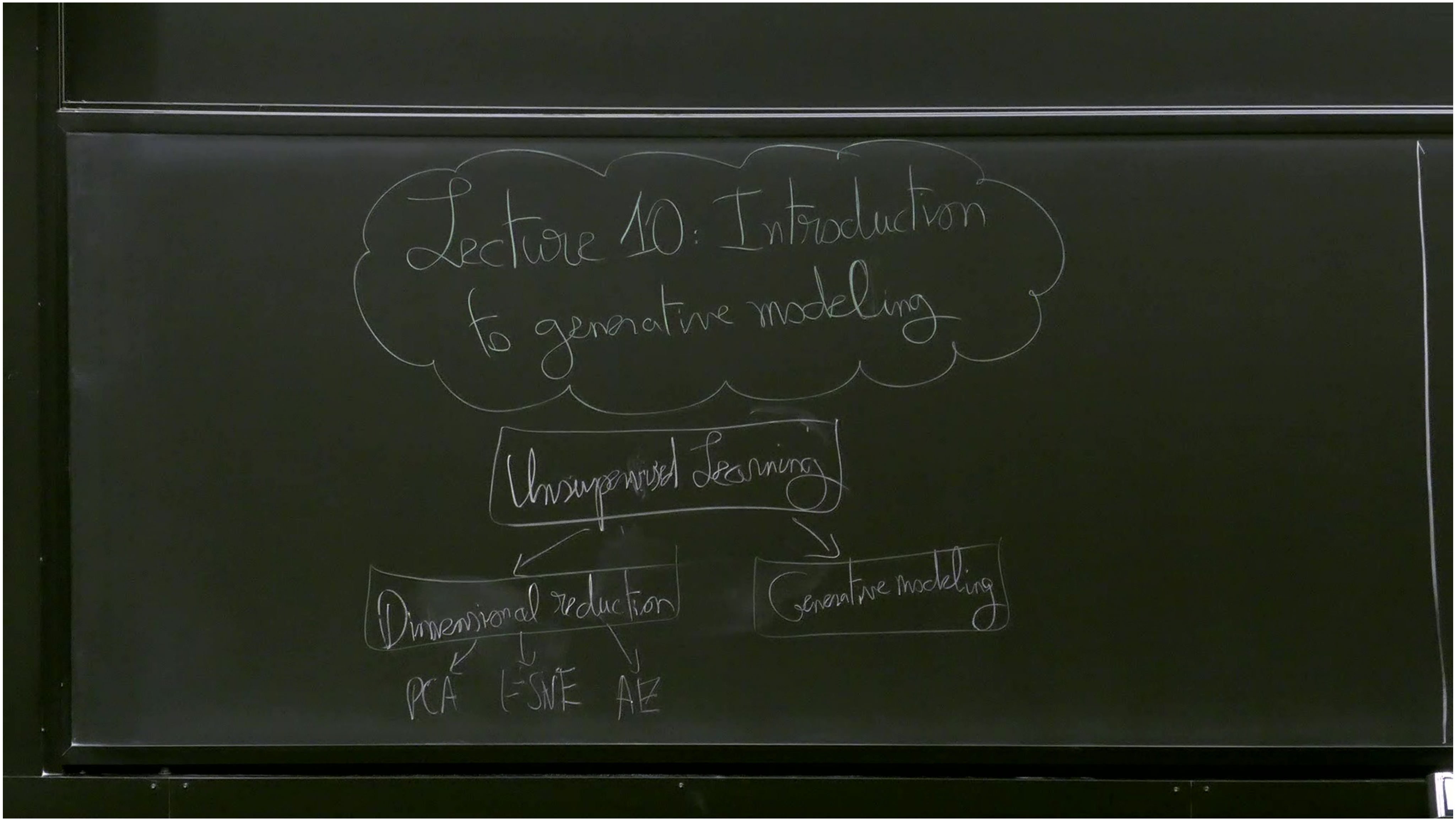
Speakers: Mohamed Hibat Allah

Collection/Series: Machine Learning (Elective), PHYS 777, February 24 - March 28, 2025

Subject: Condensed Matter, Other

Date: March 21, 2025 - 9:00 AM

URL: https://pirsa.org/25030041



Lecture 10: Introduction to generative modeling

Unsupervised Learning

Dimensional reduction

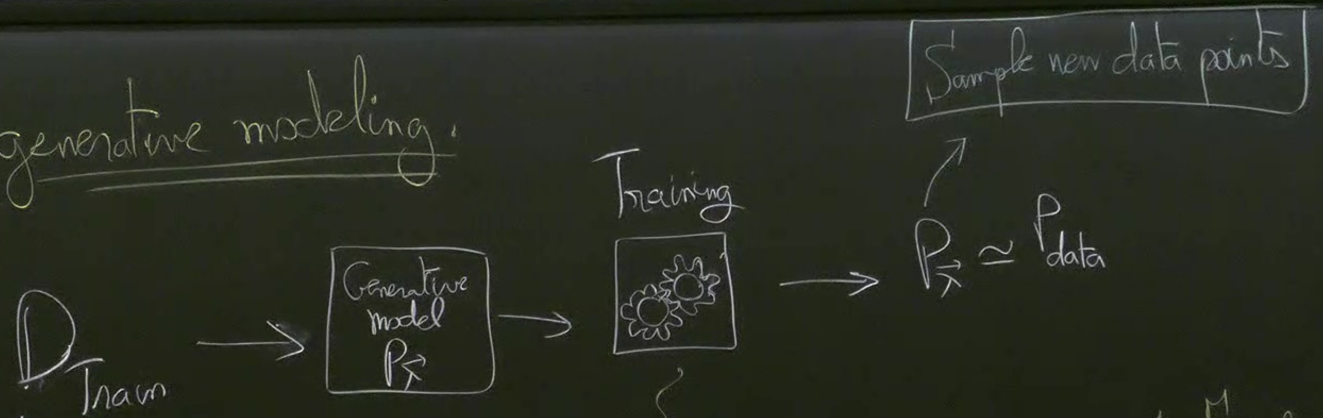
Generative modeling

PCA

t-SNE

AE

Goal of generative modeling:

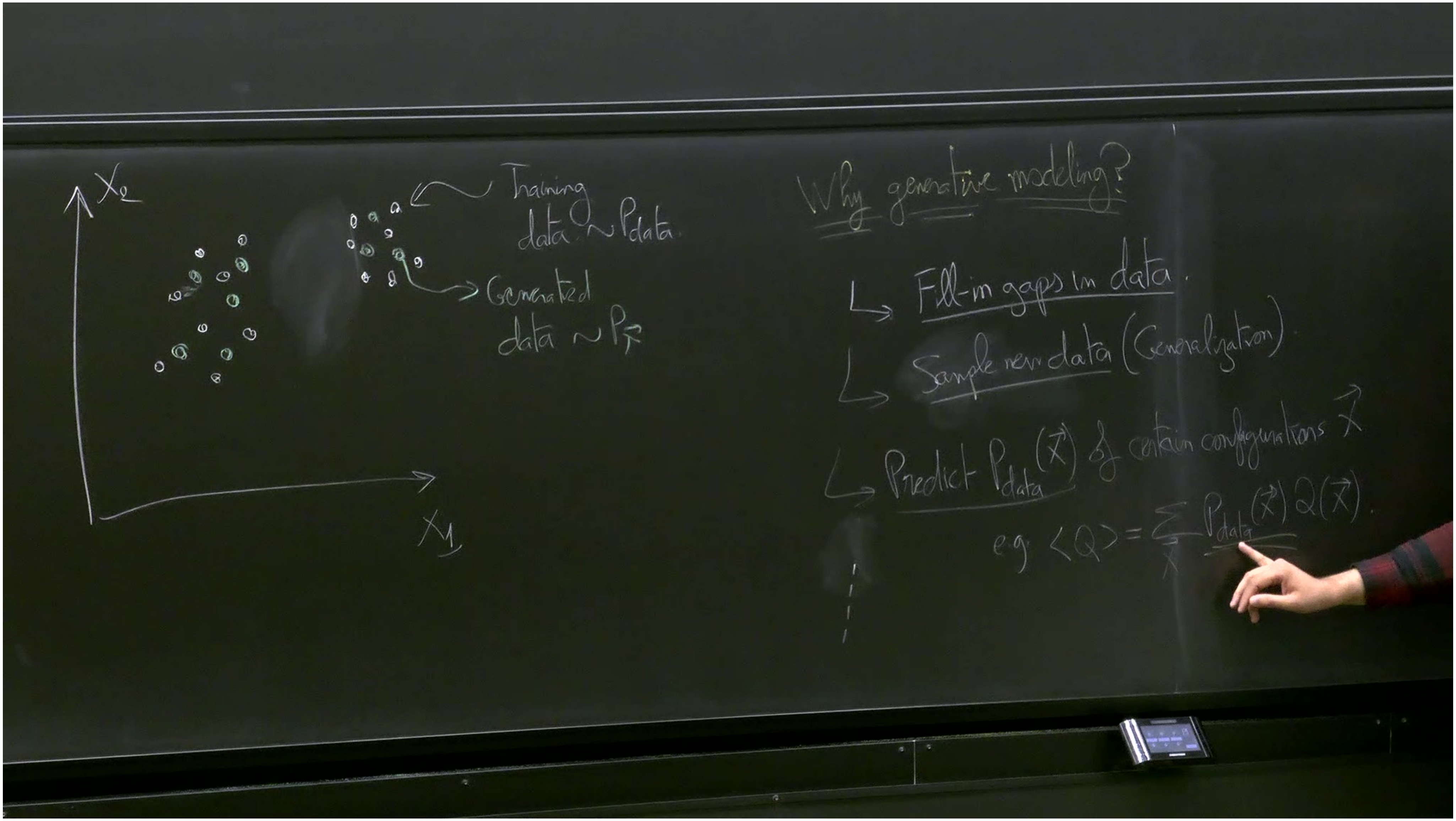


$\{\vec{x}_i\}_{i=1}^{M} \sim P_{data}$
(unknown)

$$NLL = \langle -\log(P_{\lambda}(\vec{x}_i)) \rangle_{\vec{x}_i \sim P_{data}} \simeq \frac{1}{M}\sum_{i=1}^{M} -\log(P_{\lambda}(\vec{x}_i))$$

"Numerical" convenience $\longrightarrow -\log$

Likelihood $\mathcal{L}_{\vec{\lambda}} = \prod_{i=1}^{M} P_{\vec{\lambda}}(\vec{x}_i)$

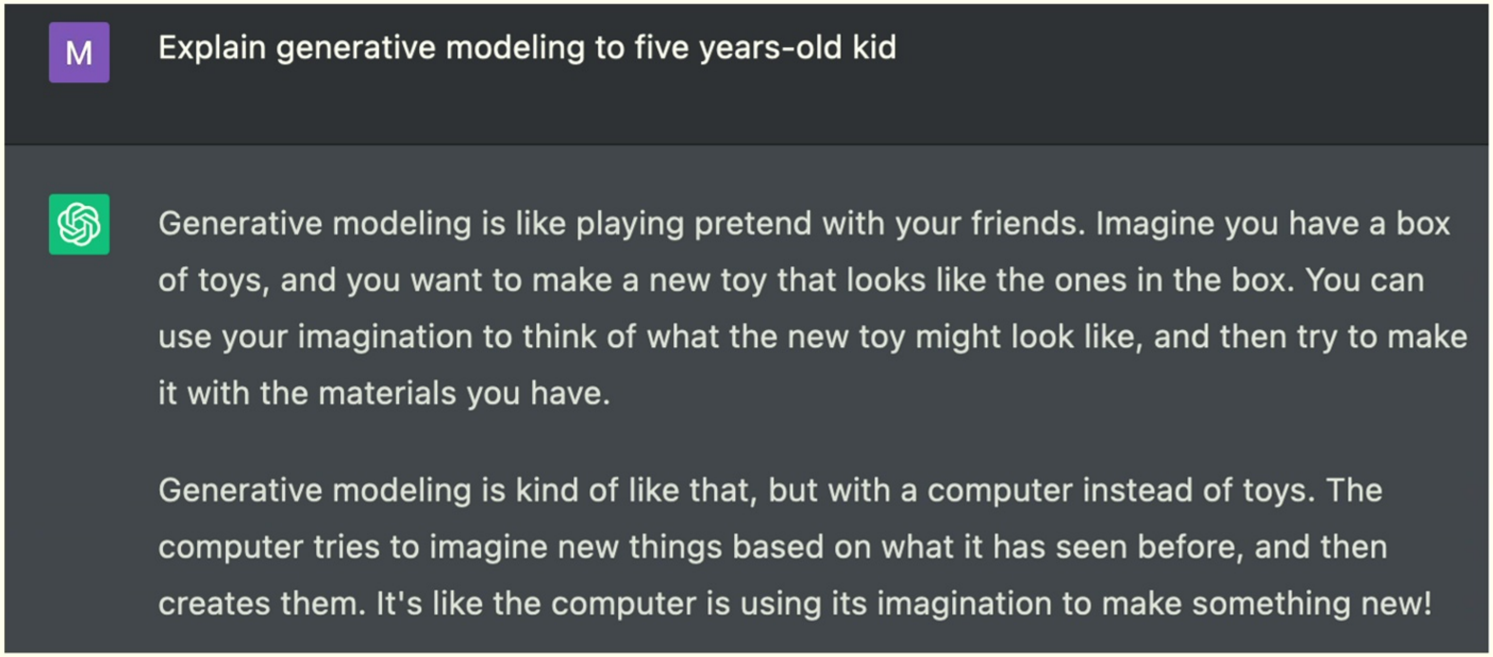

X_2
X_1
Training data $\sim P_{data}$
Generated data $\sim P_{\vec\theta}$

Why generative modeling?

- Fill-in gaps in data.
- Sample new data (Generalization).
- Predict $P_{data}(\vec{x})$ of certain configurations $\vec{x}$

e.g. $\langle Q \rangle = \sum_{\vec{x}} P_{data}(\vec{x})\, Q(\vec{x})$.

Chat-GPT

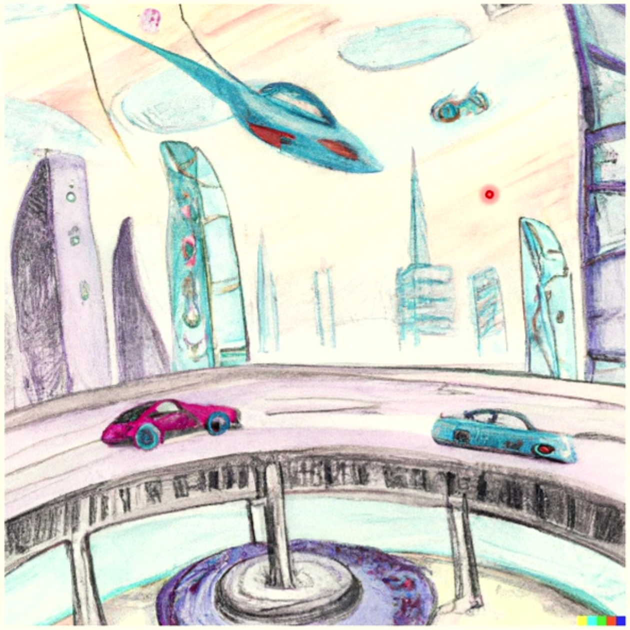


Explain generative modeling to five years-old kid

Generative modeling is like playing pretend with your friends. Imagine you have a box of toys, and you want to make a new toy that looks like the ones in the box. You can use your imagination to think of what the new toy might look like, and then try to make it with the materials you have.

Generative modeling is kind of like that, but with a computer instead of toys. The computer tries to imagine new things based on what it has seen before, and then creates them. It's like the computer is using its imagination to make something new!

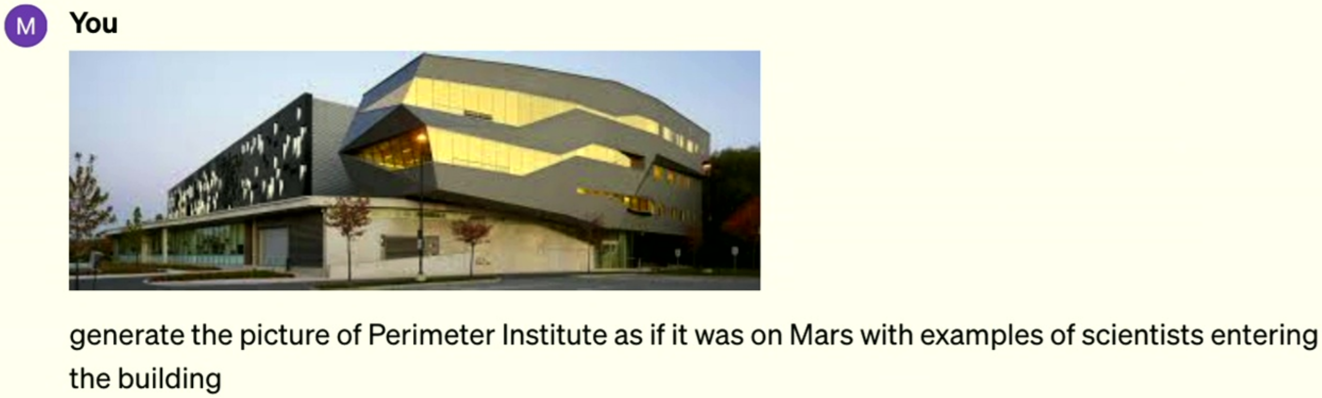
Dall-E



A pencil and watercolor drawing of a bright city in the future with flying cars

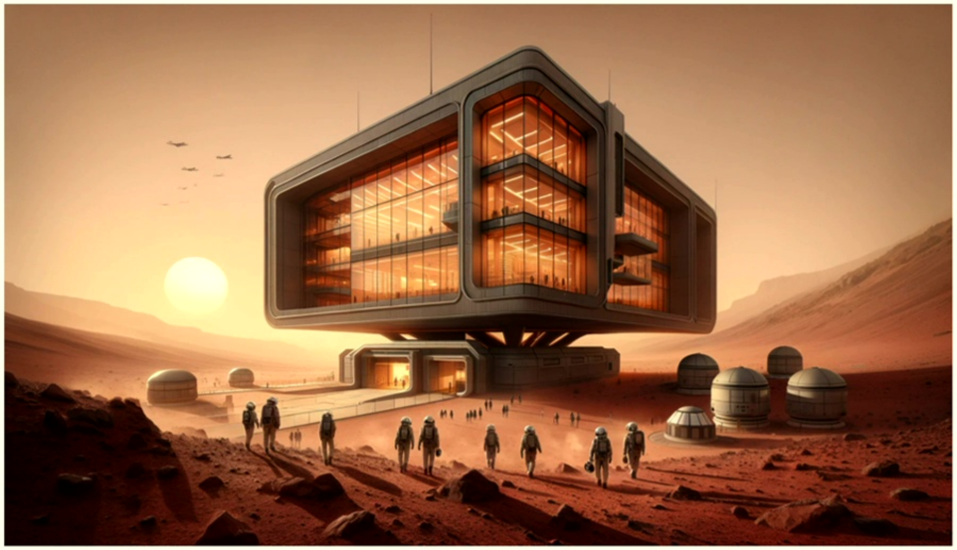
https://openai.com/product/dall-e-2

Dall-E



You

generate the picture of Perimeter Institute as if it was on Mars with examples of scientists entering the building



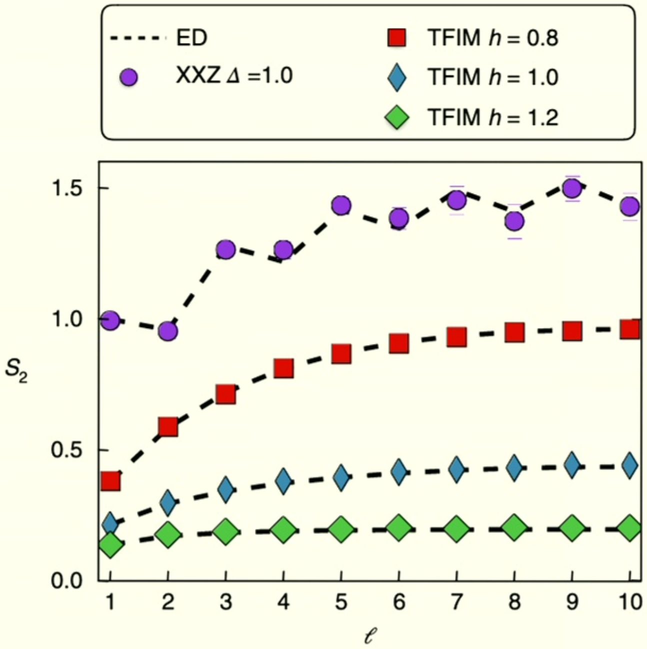
Artificial Intelligence Virtual Artist

Quantum state tomography

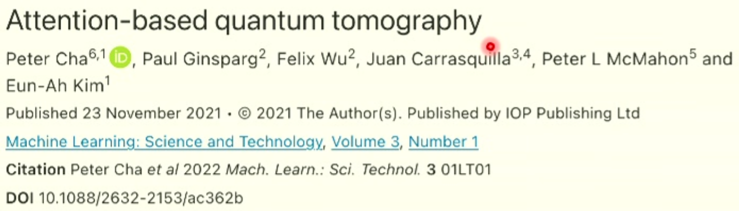
Neural-network quantum state tomography

Giacomo Torlai[1,2], Guglielmo Mazzola[3], Juan Carrasquilla[4,5], Matthias Troyer[3,6], Roger Melko[1,2] and Giuseppe Carleo[3,7*]

Starting from a **dataset in the z basis**, we can obtain **observables in x and y basis,** and also obtain challenging quantities such as the **entanglement entropy.**



Quantum state tomography



Attention-based quantum tomography

Peter Cha[6,1], Paul Ginsparg[2], Felix Wu[2], Juan Carrasquilla[3,4], Peter L McMahon[5] and Eun-Ah Kim[1]

Published 23 November 2021 • © 2021 The Author(s). Published by IOP Publishing Ltd

Machine Learning: Science and Technology, Volume 3, Number 1

Citation Peter Cha *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 01LT01

DOI 10.1088/2632-2153/ac362b

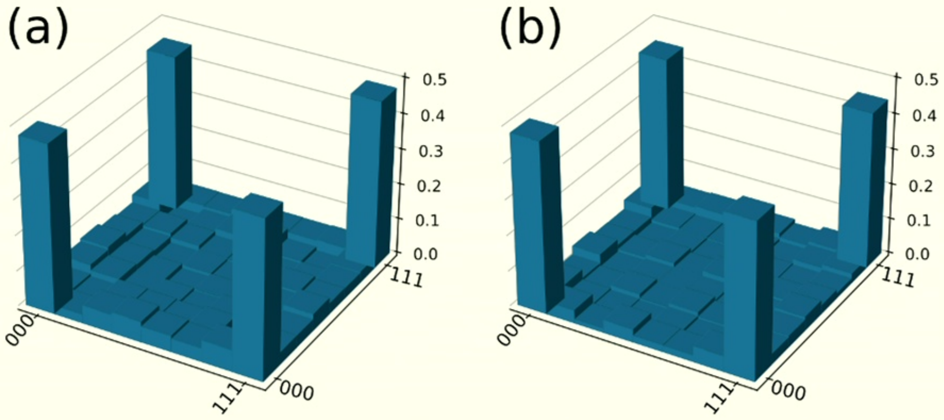
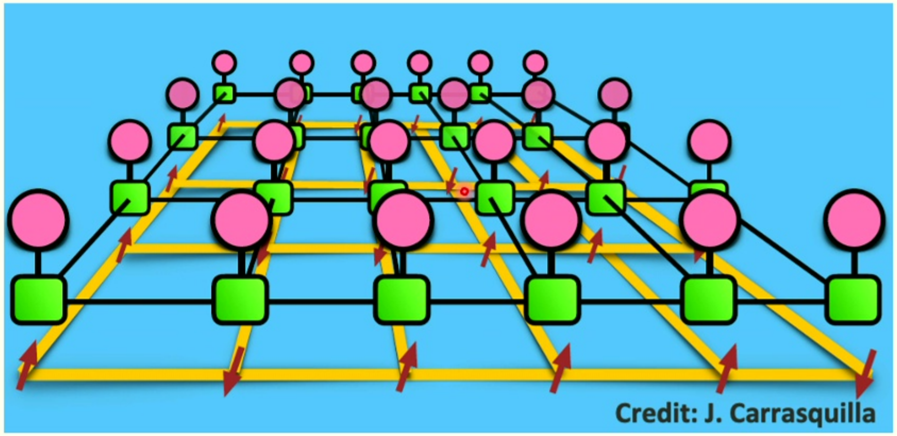


Figure 3. Benchmarking AQT (a) to MLE tomography offered by IBM's Qiskit library (b) for a noisy 3-qubit GHZ state data generated on the IBMQ_OURENSE quantum computer. Each bar represents the absolute value of a density matrix (DM) element.

Generative language modeling meets **quantum physics**

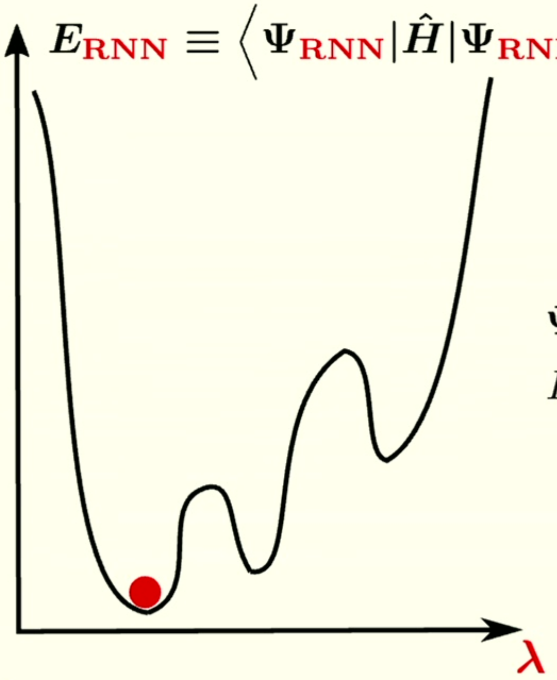
Ground states search using Recurrent Neural Networks



Credit: J. Carrasquilla

M.H, M. Ganahl, L. Hayward, R. Melko, J. Carrasquilla, **Recurrent Neural Network Wave Functions**, PRReasearch, 2020.

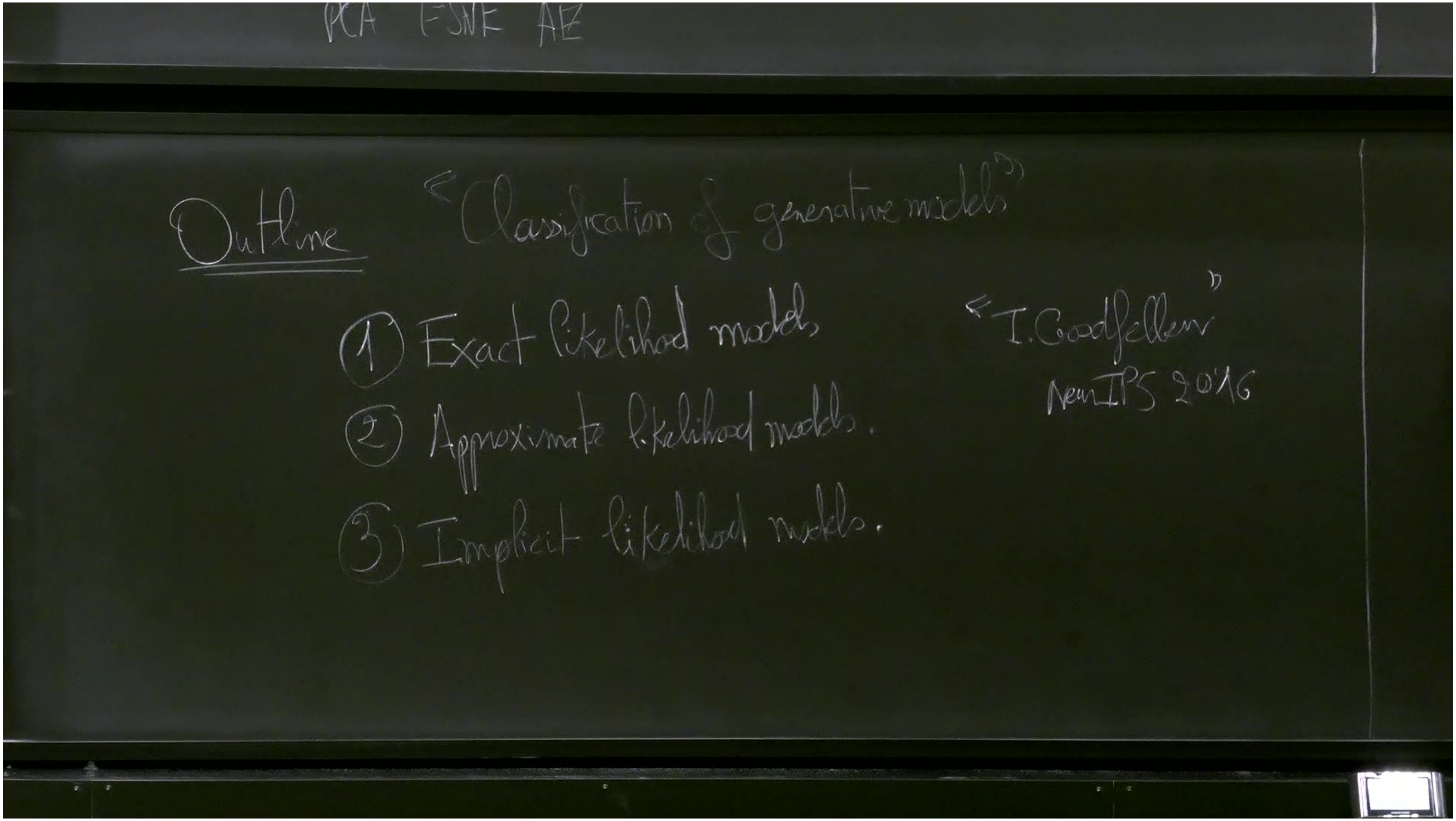
$$E_{\mathbf{RNN}} \equiv \left\langle \Psi_{\mathbf{RNN}} \middle| \hat{H} \middle| \Psi_{\mathbf{RNN}} \right\rangle$$



$$\Psi_{\mathbf{RNN}} \approx \Psi_{\boldsymbol{G}}$$

$$E_{\mathbf{RNN}} \approx E_G$$

See Lecture 12



PCA t-SNE AE

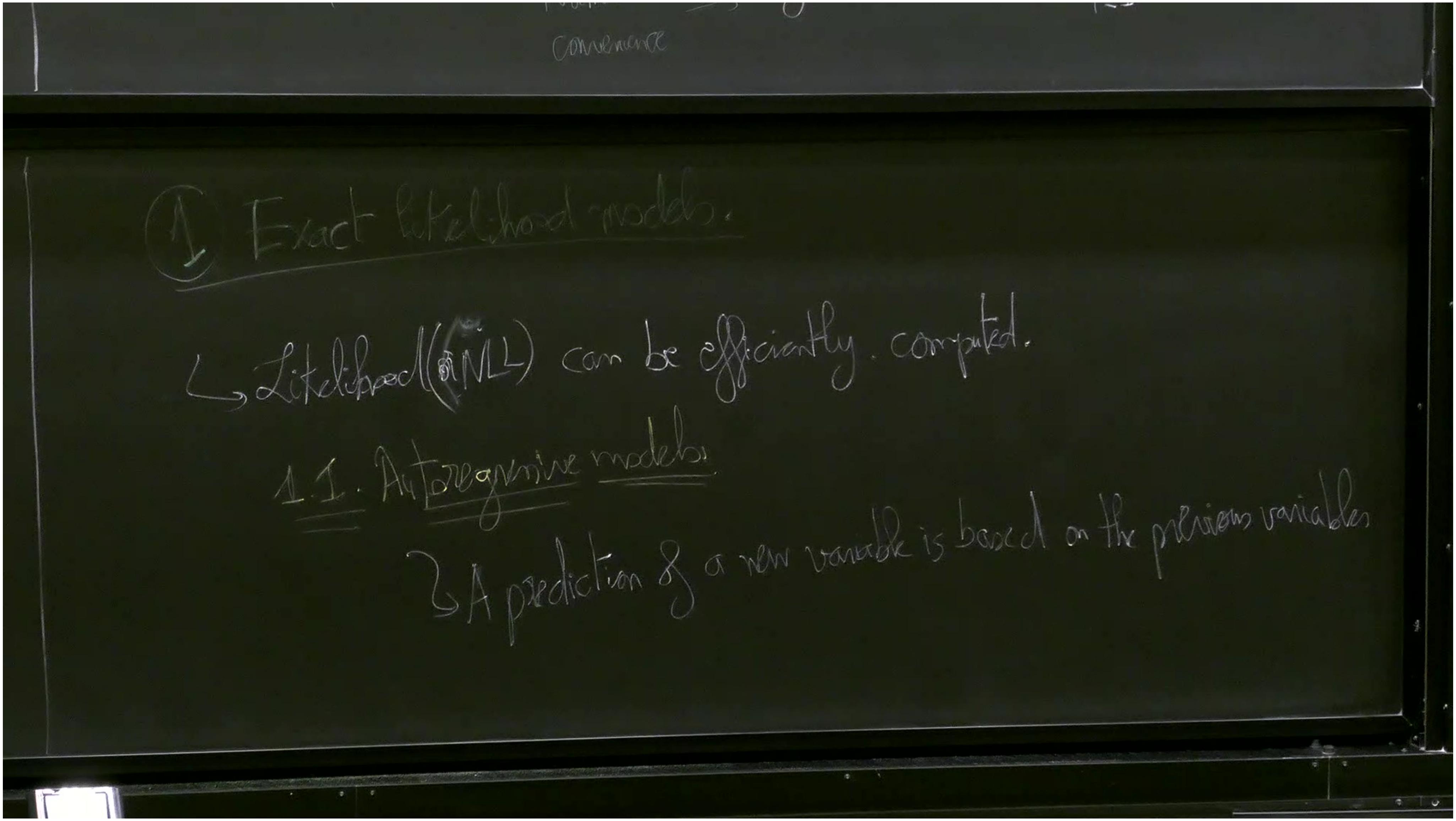
Outline "Classification of generative models"

① Exact likelihood models

② Approximate likelihood models.

③ Implicit likelihood models.

"I. Goodfellow"
NeurIPS 2016



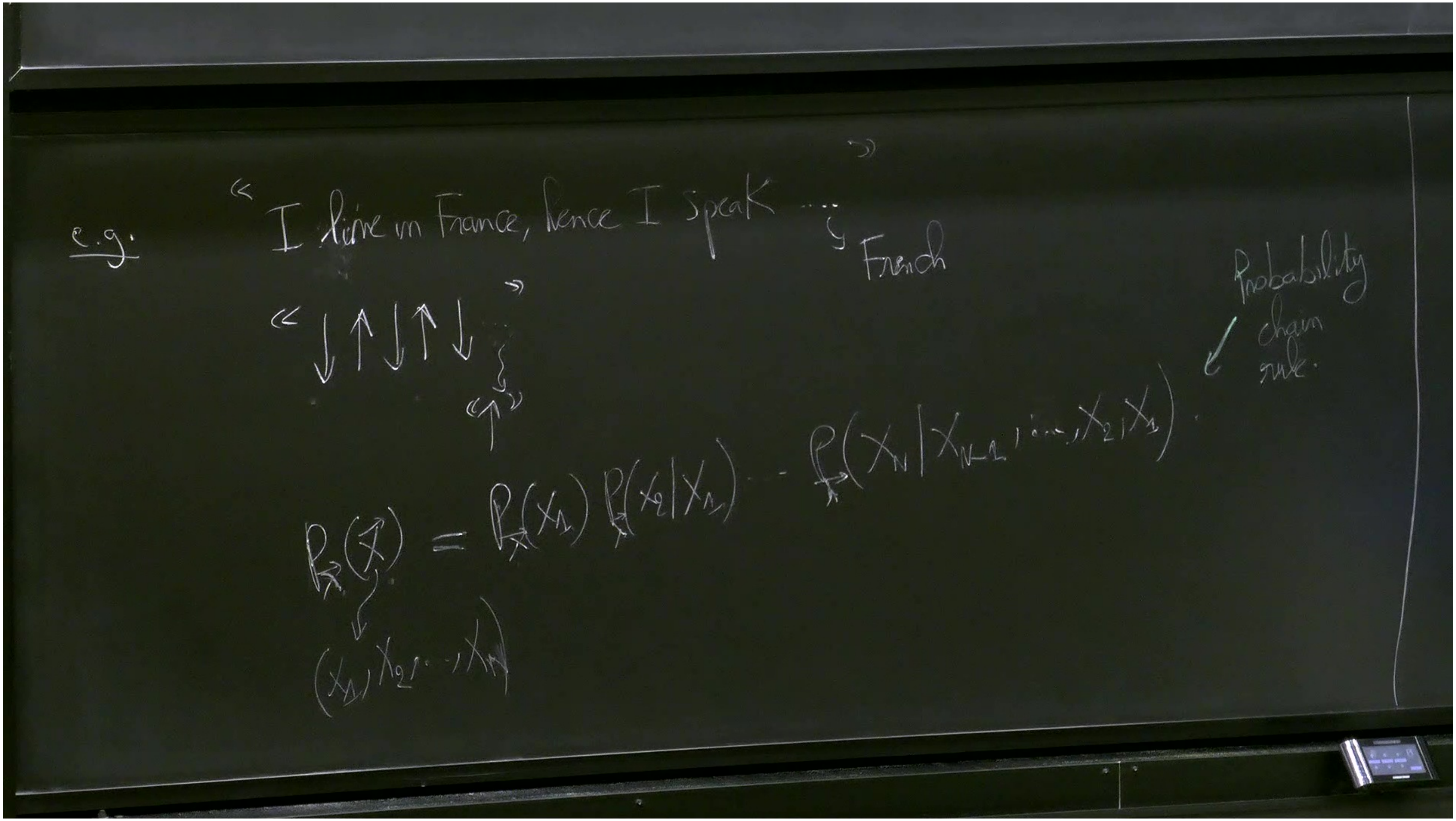
convenience

① Exact likelihood models.

↳ Likelihood (NLL) can be efficiently computed.

1.1. Autoregressive models.

↳ A prediction of a new variable is based on the previous variables



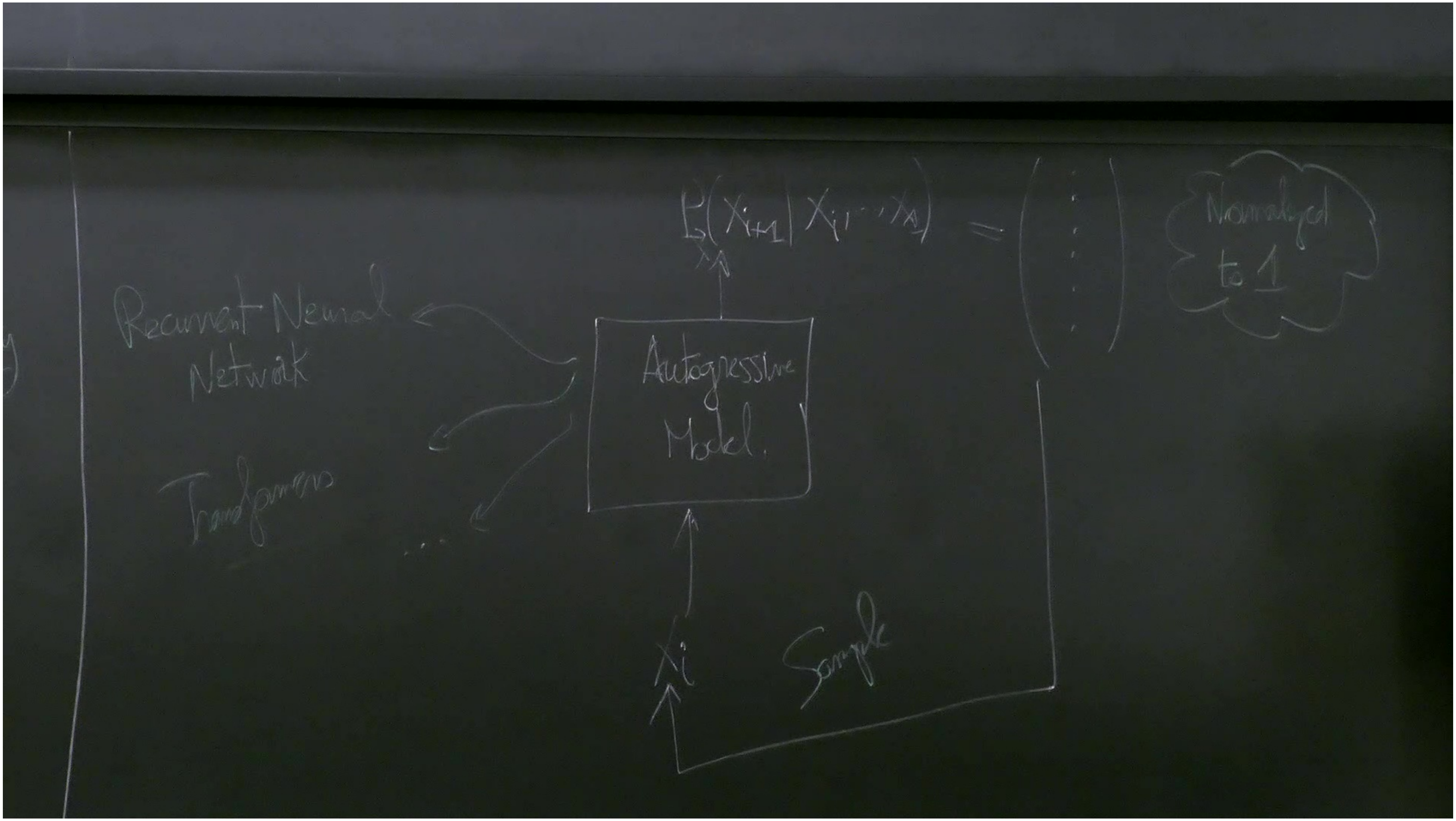
e.g. « I live in France, hence I speak ... » French

« ↓ ↑ ↓ ↑ ↓ ... » « ↑ »

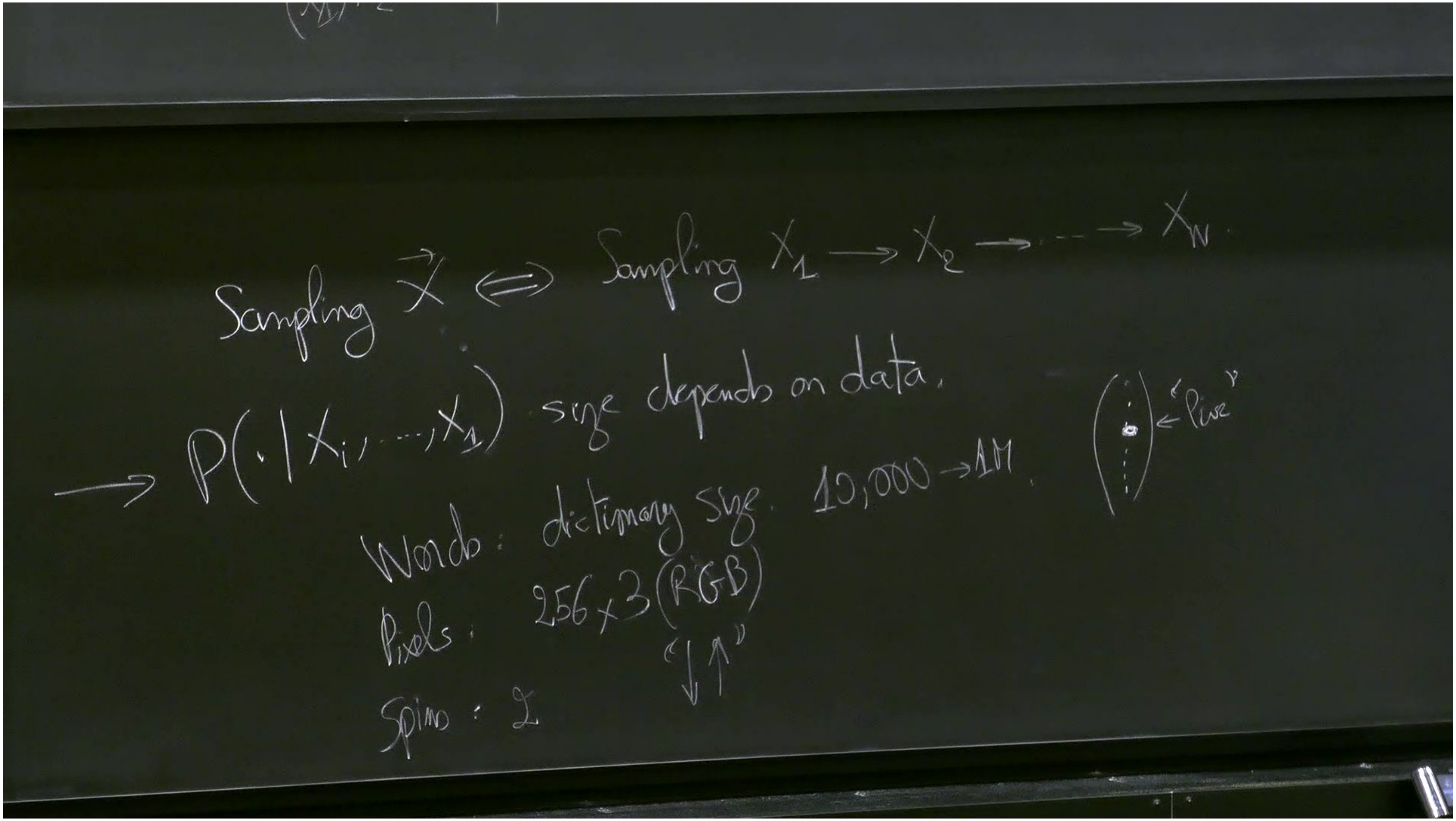
$$P_{\vec{X}}(\vec{X}) = P_{\vec{X}}(X_1)\, P_{\vec{X}}(X_2|X_1) \cdots P_{\vec{X}}(X_N|X_{N-1},\dots,X_2,X_1)$$

$(X_1, X_2, \dots, X_N)$

Probability chain rule.



$$P_\lambda(x_{i+1}|x_i,\dots,x_1) = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$$



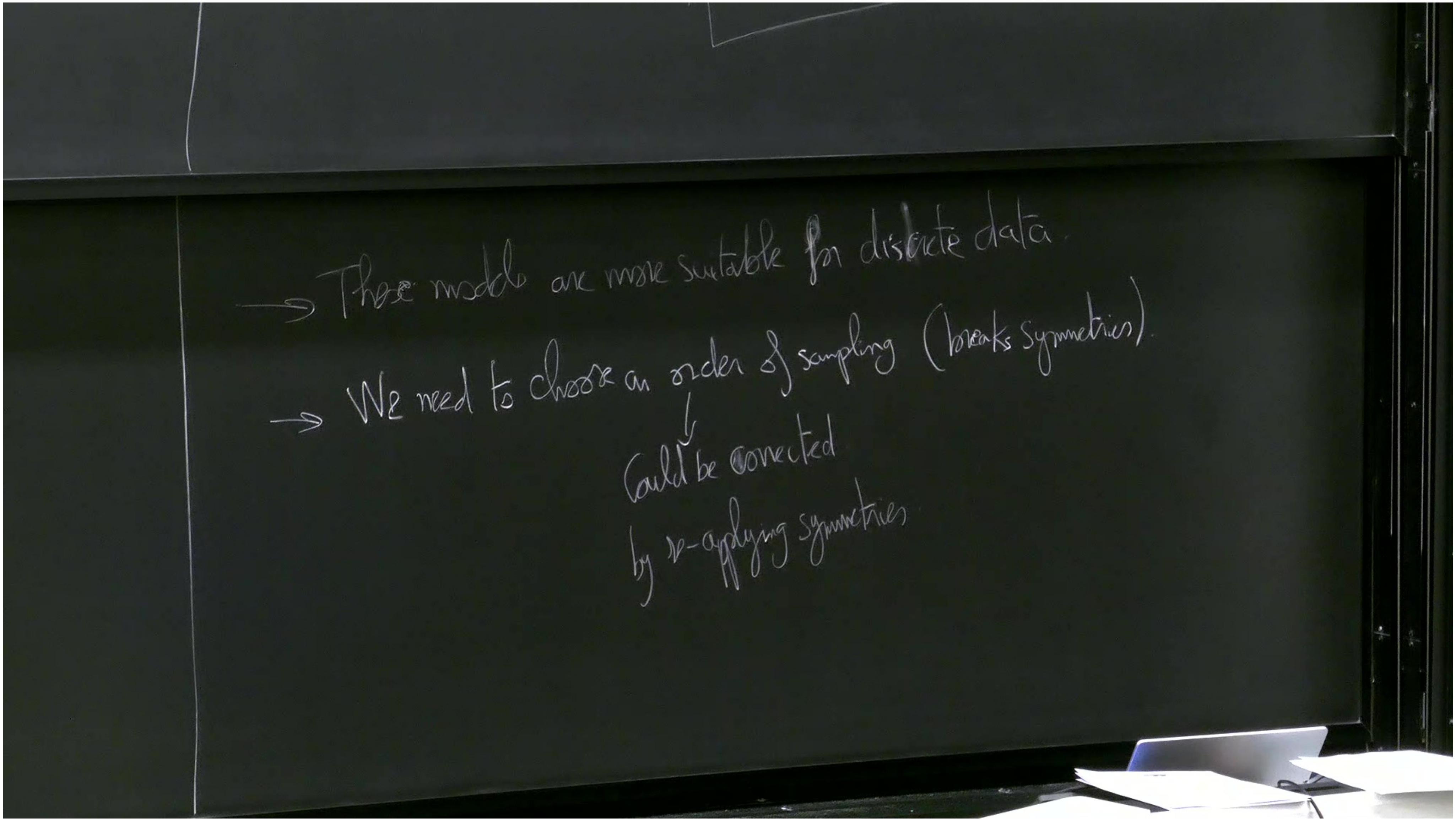
Sampling $\vec{X} \Longleftrightarrow$ Sampling $X_1 \longrightarrow X_2 \longrightarrow \cdots \longrightarrow X_N$.

$\longrightarrow P(\cdot \,|\, X_i, \dots, X_1)$ size depends on data.

Words: dictionary size. 10,000 → 1M.

Pixels: 256×3 (RGB)

Spins: 2 "↓↑"

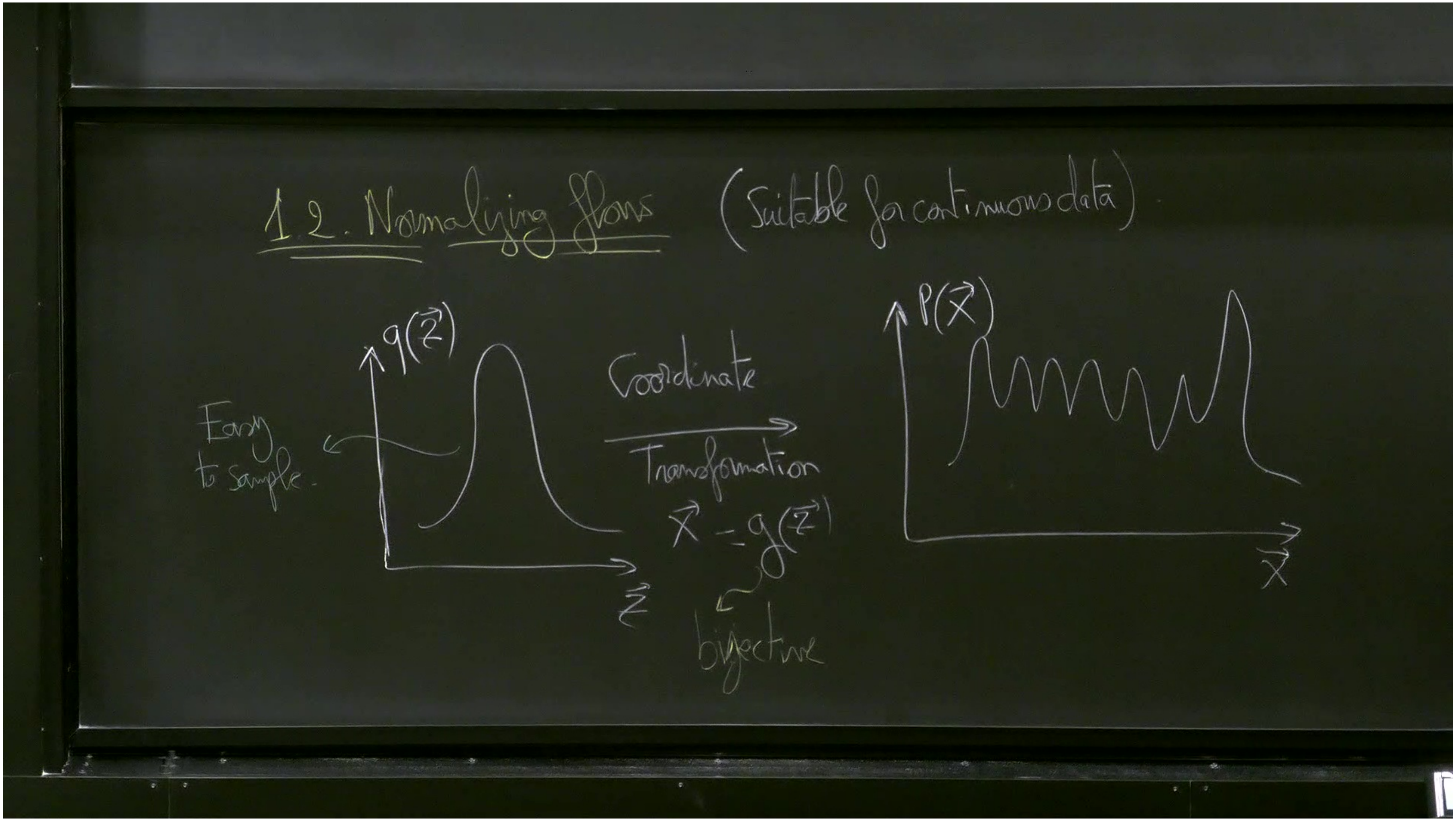


→ These models are more suitable for discrete data.

→ We need to choose an order of sampling (breaks symmetries).

↓

Could be corrected
by re-applying symmetries.



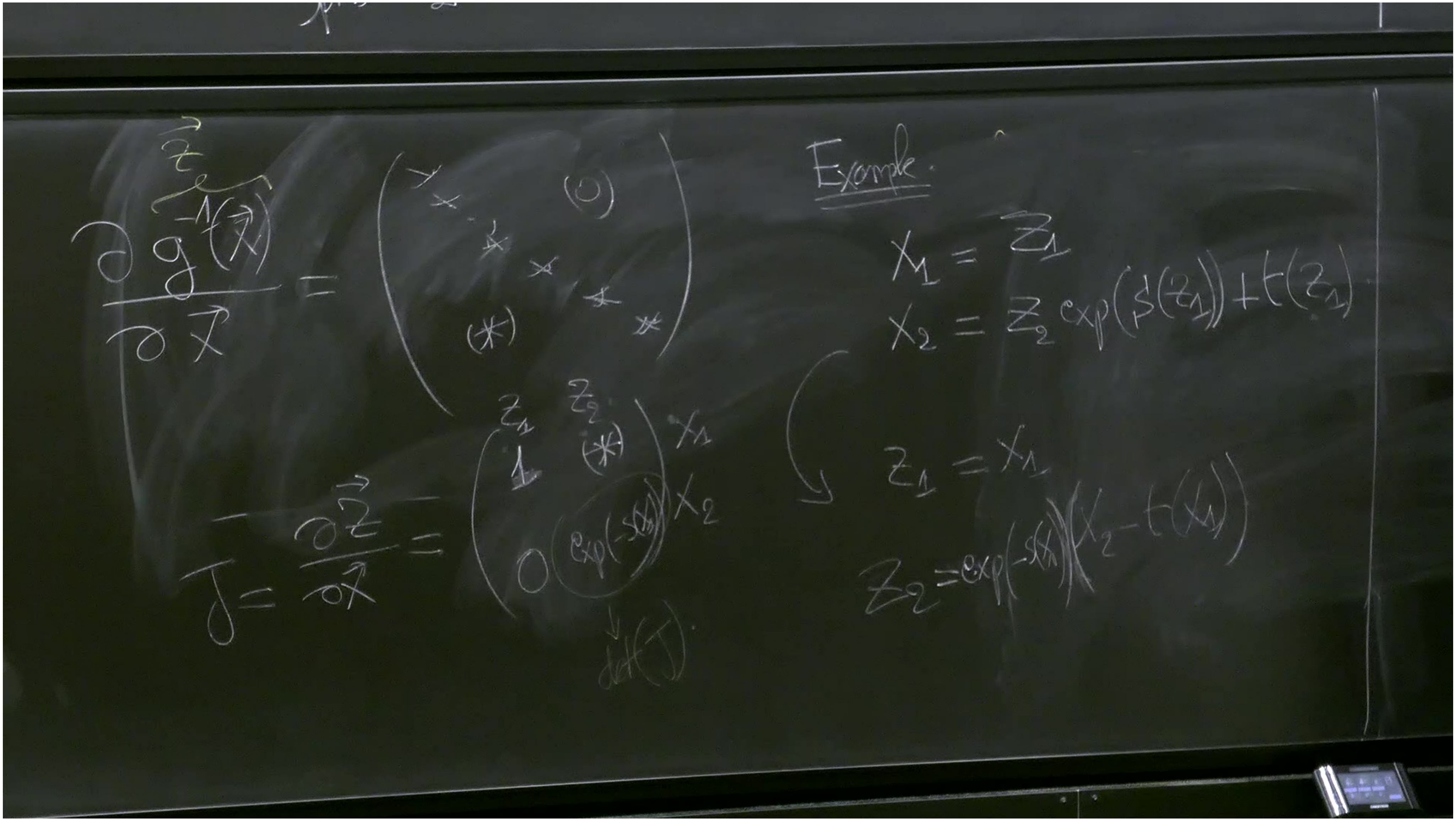
1.2. Normalizing flows (Suitable for continuous data)

$$\int q(z)\,dz = 1 \xleftrightarrow[\text{of Variable}]{\text{Change}} \int q\big(\underbrace{g^{-1}(x)}_{z}\big)\left|\frac{dz}{dx}\right| dx = 1$$

$$p(x) = q(z)\cdot\left|\frac{dz}{dx}\right|.$$

For multiple variables:

$$p(\vec{x}) = q\big(\overset{g^{-1}(\vec{x})}{\vec{z}}\big)\cdot\left|\overset{\text{Jacobian}}{\det\left(\frac{\partial\vec{z}}{\partial\vec{x}}\right)}\right|$$



$$\frac{\partial \overbrace{g^{-1}(\vec{x})}^{\vec{z}}}{\partial \vec{x}} = \begin{pmatrix} \times & & & & (0) \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ (*) & & & & \times \end{pmatrix}$$

$$J = \frac{\partial \vec{z}}{\partial \vec{x}} = \begin{array}{cc|c} z_1 & z_2 & \\ 1 & (*) & x_1 \\ 0 & \exp(-s(x_1)) & x_2 \end{array} \quad \exp(-s(x_1)) \rightarrow \det(J)$$

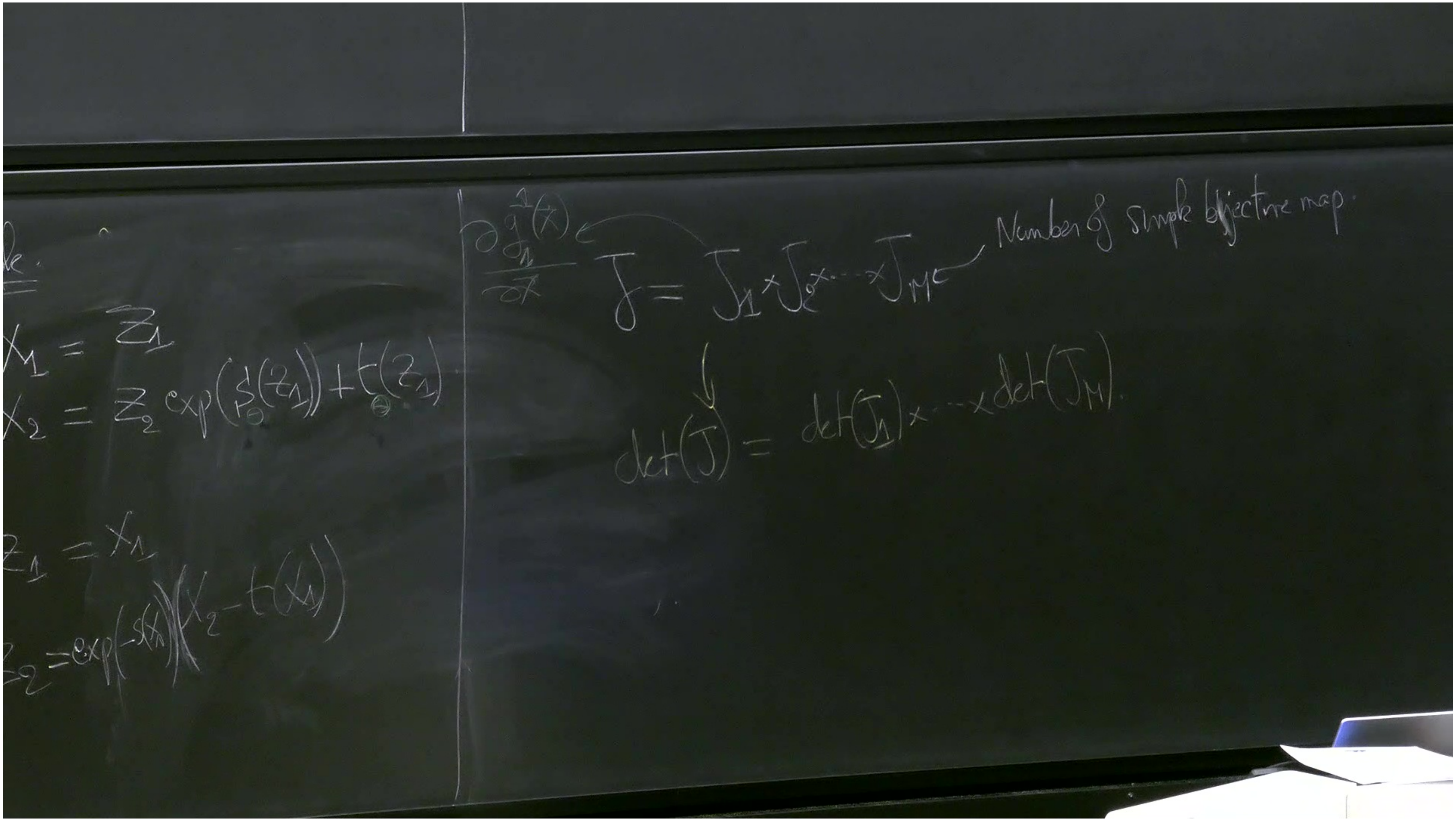
Example.

$$x_1 = z_1$$

$$x_2 = z_2 \exp(s(z_1)) + t(z_1)$$

$$z_1 = x_1$$

$$z_2 = \exp(-s(x_1))(x_2 - t(x_1))$$



le.

$X_1 = Z_1$

$X_2 = Z_2 \exp(s_\theta(Z_1)) + t_\theta(Z_1)$

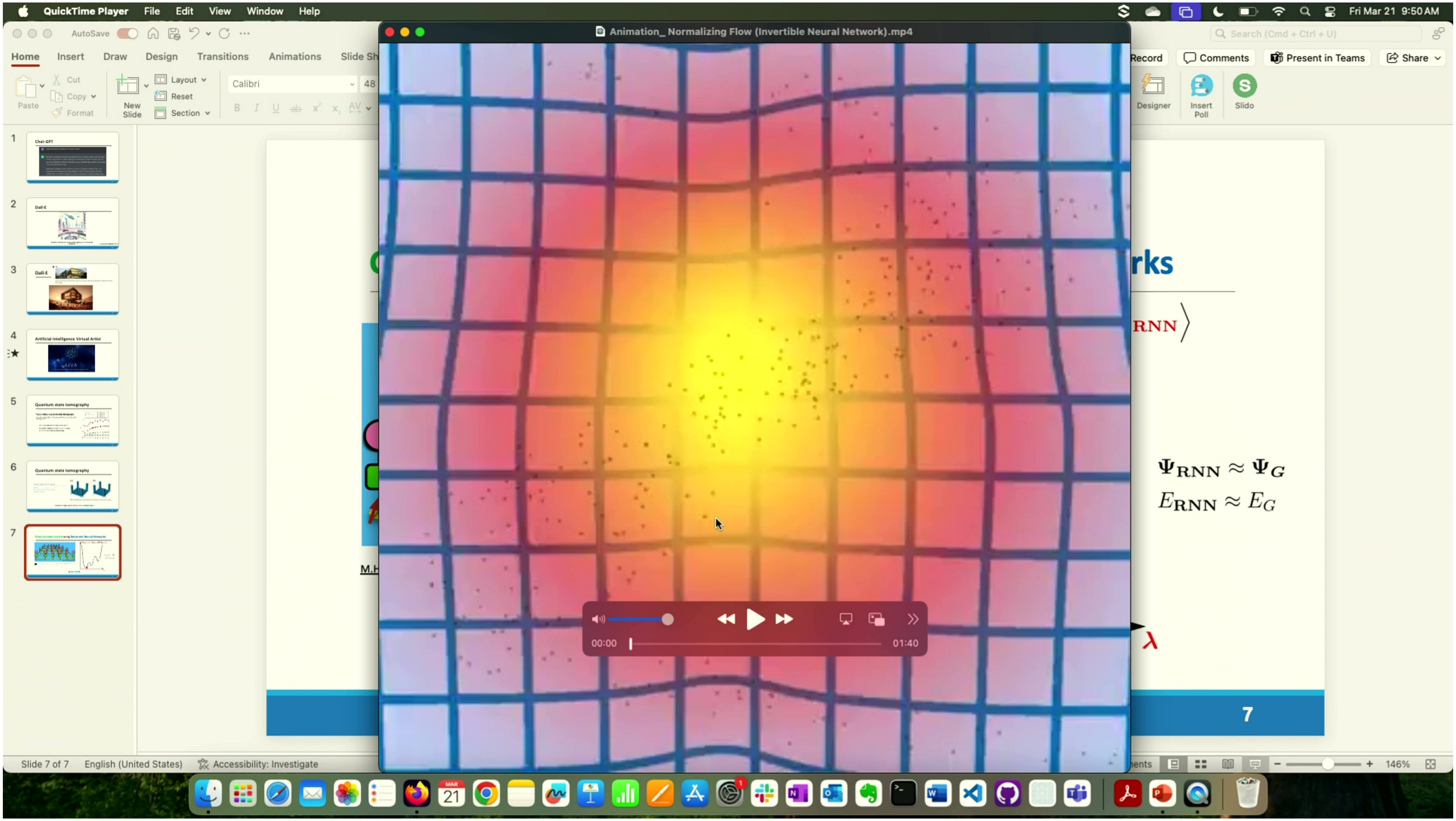
$Z_1 = X_1$

$Z_2 = \exp(-s(X_1))(X_2 - t(X_1))$

$\frac{\partial g_1^{-1}(x)}{\partial x}$

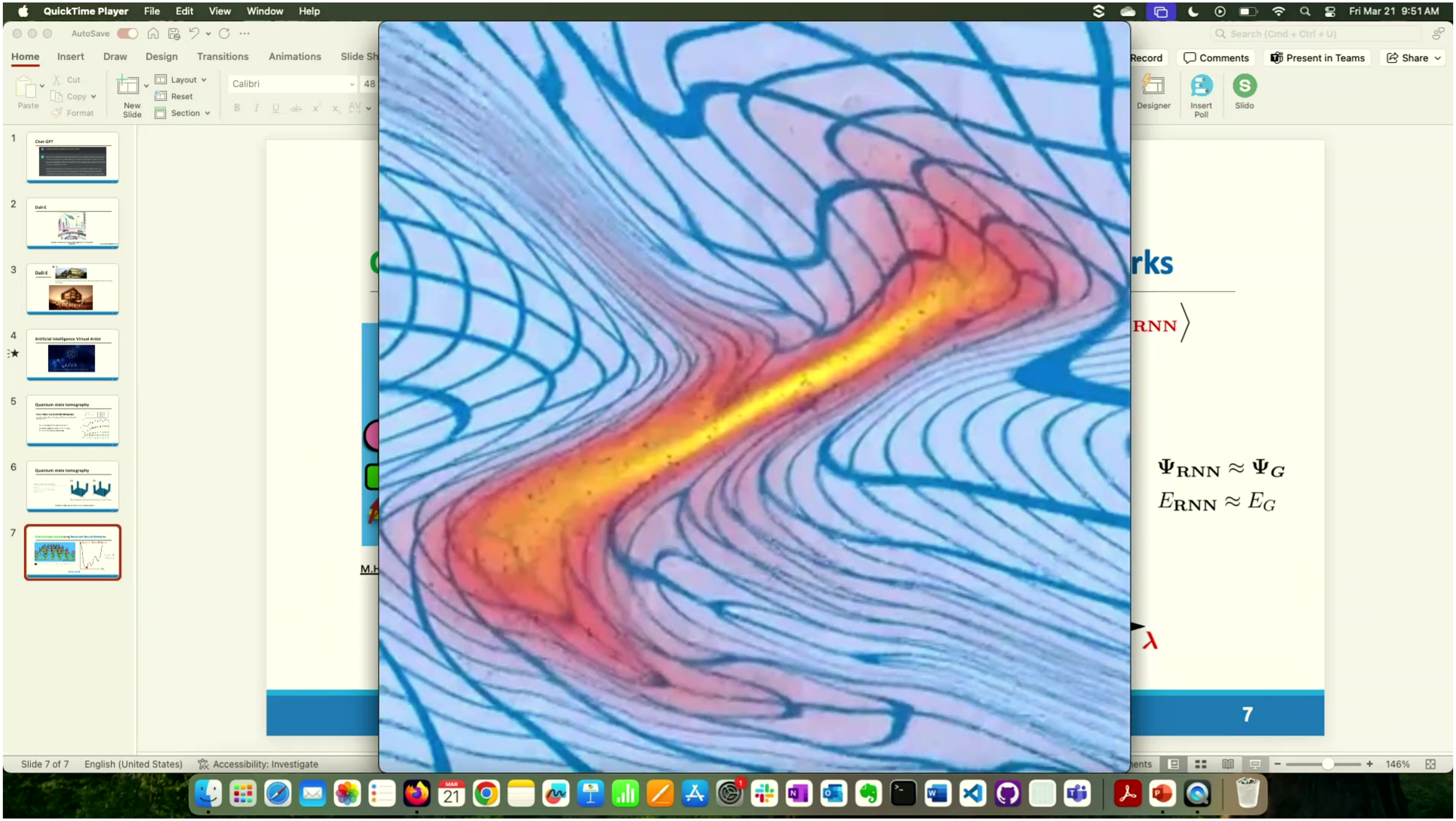
$J = J_1 \times J_2 \times \cdots \times J_M$ Number of simple bijective map.

$\det(J) = \det(J_1) \times \cdots \times \det(J_M).$



$$\Psi_{\text{RNN}} \approx \Psi_G$$

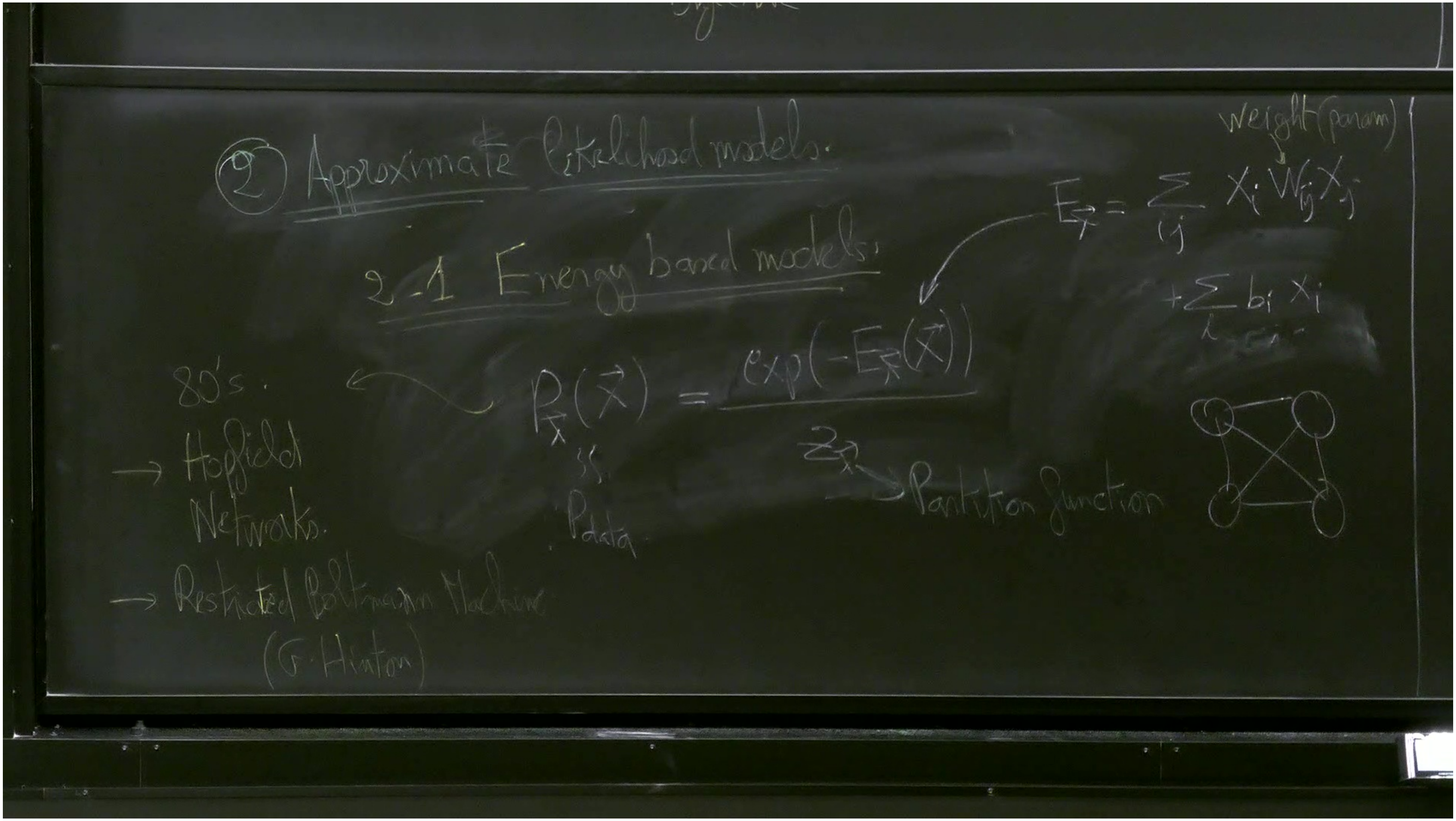
$$E_{\text{RNN}} \approx E_G$$



$\Psi_{\mathrm{RNN}} \approx \Psi_G$

$E_{\mathrm{RNN}} \approx E_G$

λ



② Approximate likelihood models.

2-1 Energy based models.

80's.

→ Hopfield Networks.

→ Restricted Boltzmann Machine (G. Hinton)

$$P_{\vec{x}}(\vec{x}) = \frac{\exp(-E_{\vec{x}}(\vec{x}))}{Z_{\vec{x}}}$$

P_{data}

$Z_{\vec{x}} \rightarrow$ Partition function

weight (param)

$$E_{\vec{x}} = \sum_{ij} x_i W_{ij} x_j + \sum_i b_i x_i$$

$$P(\vec{X}) = q(Z) \cdot \left|\det\left(\frac{}{\partial \vec{X}}\right)\right|$$

$$Z_{\vec{X}} = \sum_{\vec{X}} \exp\left(-E_{\vec{X}}(\vec{X})\right).$$ "Intractable"

$$\vec{X} = \underbrace{(X_1, \dots, X_N)}_{2^N}$$

$$\frac{P(\vec{X}_2)}{P(\vec{X}_1)} = \exp\left(-\overbrace{\left(E_{\vec{X}}(\vec{X}_2) - E_{\vec{X}}(\vec{X}_1)\right)}^{\text{Tractable}}\right).$$

We can use. Metropolis sampling (for instance)